Symmetry Based Semantic Analysis of Engineering Drawings

Thomas C. Henderson, Narong Boonsirisumpun, and Anshul Joshi
University of Utah, SLC, UT, USA;
tch at cs.utah.edu

***Abstract*— Engineering drawings have posed significant challenges to image analysis for many decades. The goal is to take images of scanned engineering drawings and interpret them so as to understand their contents (e.g., characters, digits, line segments, box segments etc.). This is known as *semantic analysis*. We propose a new approach here which takes advantage of the man-made nature of drawings: there is a tremendous amount of symmetry. We exploit this insight to enhance our previously reported system, the Non-Deterministic Agent System (NDAS), with symmetry-based analysis tools. Agents work independently but use each others results to produce the final result (e.g., form segmentation, character analysis, structural analysis, boundary segmentation, etc.). We use the wreath product representation both to characterize symmetry as well as to structure a Bayesian network model of the uncertainty. This approach permits wide application to perform semantic analysis of engineering drawings.**

I. Introduction and Background

Figure 1 shows an example of a complex engineering drawing and the kinds of semantic information which must be derived from the image. Much work has been done in this area, and we refer the reader to [2] for a comprehensive survey and detailed review of the state-of-the-art. We proposed the **N**on**D**eterministic **A**gent **S**ystem (NDAS) [3], which is a set of nondeterministic image analysis agents that use a blackboard to communicate. One weakness of the system concerns the extraction of robust low-level geometric primitives: straight line segments, arcs, etc. We propose here a new approach to the extraction of shape based on the use of specific symmetries in geometric forms. The particular geometric entities to be found include: lines, boxes, circles, polygons, and corners. We propose to determine the standard coordinate axes of the image as well (which way a person sees as up). Note that linear structure may also be used to detect lines of text in the image, provided that there is sufficient text.

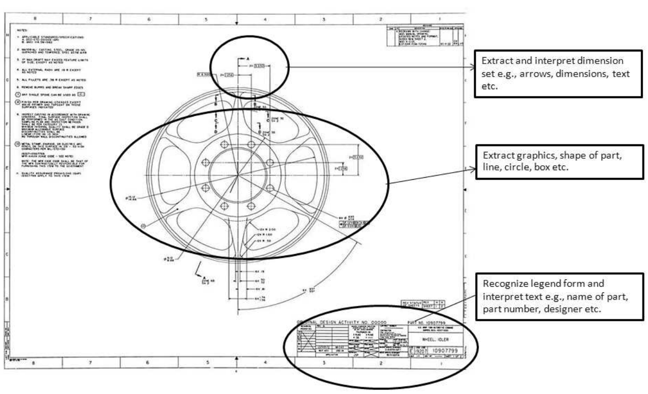

Fig. 1. An Example Engineering Drawing and the Kinds of Semantic Analysis Desired.

II. Symmetry Theory and Wreath Products

We have been studying the role of symmetry in robot cognition (see [4]), and follow Leyton's basic approach. He proposed a generative model of shape [6] based on the *wreath product* group. (Also, see [7], [8] for a discussion of the key issue of invariance as a way to detect regularities in geometric objects.) The wreath product, denoted $F \wr C$, is defined as the semi-direct product of two groups, F and C, where C is the *control* (permutation) group which acts on F the *fiber* group. More formally:

$$F \wr C \equiv \prod_{i=1}^{n} F_i \rtimes C$$

where $\rtimes$ is the semi-direct product of n copies of F with C. C is generally a permutation group with the permutations applied to the copies of F. The key notion is that C is the control group that acts to transform the fiber group elements onto each other.

We apply this idea directly to low-level image analysis. As a simple explanation:

- *the translation symmetry group – denoted by $\Re$ (1D)*: the invariance of pixel sets under translation defines a straight line segment.
- *the rotation symmetry group (2D) – denoted by* $O(2)$: the invariance of pixel sets under rotation defines a circle (and with a small modification allows description of ellipses).
- *the reflection symmetry group – denoted by Z_2 (2D)*: the invariance of a set of pixels under reflection about a line in the plane describes bilateral symmetry in 2D.

We show that this approach significantly improves the segmentation of low-level geometric primitives.

A. Group Theory

Group theory is the mathematical formalization of symmetry. A group G is a set, S, along with a binary operator $\star$ defined as the ordered pair $(S, \star)$ which satisfies the following axioms:

- **Closure**: $\star : S \times S \rightarrow S$.
- **Identity**: $\exists\ \varepsilon \in S \mid \forall a \in S, a \star e = e \star a = a$
- **Inverse**: $\forall a \in S, \exists\ a^{-1} \in S \mid, a \star a^{-1} = a^{-1} \star a = \varepsilon$
- **Associativity**: $(a \star b) \star c = a \star (b \star c), \forall a, b, c \in S$

B. Wreath Product

Leyton ([6]) defines a wreath product group (WPG) as an extension of its normal subgroup (the direct product of the group) and a permutation group. WPGs aim to maximize:

- **transfer of action**: actions can be applied to new situations based on previous situations, and
- **recoverability**: ability to give causal explanation to effects.

A WPG is composed of an upper group, called the *control group* (C), acting on a lower group, the *fiber group* (F), that moves the F around (onto copies of its elements) and represents the actions that are transferred. The *fiber group* is related to the *control group* by a semi-direct product.

The basic groups of interest can be described as follows:

1) *an infinite line in 2D*, written as $\{e\} \wr \Re$: represents a translation. This breaks down as:
   - $\{e\}$: this represents a point (in the plane) by using the group consisting of just the identity element (e). The operator ($\star$) can be viewed as the identity transform in the plane (I). In any implementation, it is necessary to augment this with the actual coordinates of the point with respect to some coordinate system.
   - $\Re$: this represents all continuous translations (i.e., the real number line). Again, it is necessary to augment this with information which determines the line (e.g., a direction). $\Re$ is a 1-parameter Lie group (i.e., the set is compact and continuous).
   - *the uncountable nature of* $\Re$: since the real numbers are uncountable, the fiber group consists of the direct product of an uncountable number of points; however, since we deal with digital images, we have a discrete number of pixels to indicate.
2) *a finite length line segment*, written as $\{e\} \wr \mathcal{Z}_2 \wr \Re$: represents a finite line segment by adding a characteristic group ($\mathcal{Z}_2$) to select the point or not along the infinite translation represented by $\Re$. The digital nature of images means that a variety of options are available to represent this (e.g., two endpoints, a list of points, the parameterized equation of a line with the extreme values of the parameter specified, etc.).
3) *a circle*, written as $O(2)$: this represents the group of rotations about some point in the plane. It is a 1-parameter Lie group. This must be augmented by a center point location and radius.
4) *a reflection about a line (in 2D)*, written as $\mathcal{Z}_2$: this represents a reflection because it is a 2-element group with just the identity and the reflection; note that the reflection of the reflection is the identity. This must be augmented with a line of reflection.

The aspect of interest here is that the wreath product defines shape by means of a transformation of a point set. The control group (the group to the right of the operator) tells how to transform the elements on the left. This gives a generative model. E.g., to make a finite line segment, start with a point in the plane and translate it in a certain direction for a certain amount.

Figure 2 demonstrates the idea that a point is translated by the control group, $\Re$.

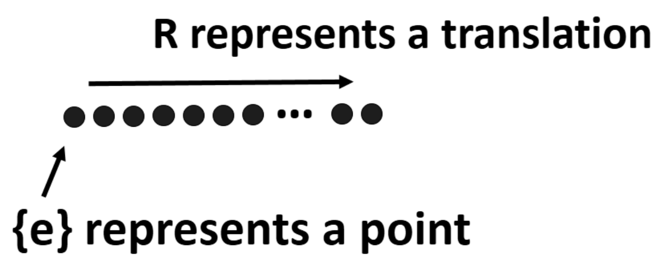

Fig. 2. Example of Control Flow in a Wreath Product; e, the identity group which represents a point is acted on by $\Re$ to obtain the an infinite line.

A more complicated example is that of a rectangular shape which can be represented as: $\{e\} \wr \mathcal{Z}_2 \wr \Re \wr \mathcal{Z}_2 \times \mathcal{Z}_2$, where $\{e\}$ represents a point, the first $\mathcal{Z}_2$ is the characteristic selection group which indicates which points are selected, $\Re$ is the translation group, and the last group is the cross product of two cyclic groups of order 2, representing reflections about the two lines which cut the rectangle across the two pairs of sides. (See [4], [6] for more details.)

III. Detecting Symmetries in Images

A. Translation

Following the notion that translation symmetry arises from the (straight line) motion of a point, we have developed the following algorithm:

Algorithm *Find_Translation*

On input: im – a binary image
On output: θ – the direction of motion
d – the translation distance along θ direction

$\forall \bar{p}$ =(row,col)
$s \leftarrow$ range scan centered at $\bar{p}$
$\theta \leftarrow$ direction of greatest possible translation
$d \leftarrow s(\theta)$

The range scan is performed at a sub-pixel level and returns the distance to the background from the given pixel in one degree steps. Figure 3 shows an image and the range scan at the indicated pixel. Figure 4 shows part of an engineering drawing (upper left), an arrow plot indicating the direction of translation at each foreground pixel (upper right), a close-up on the letter 'A' (lower left), and a close-up of two crossing lines (lower right). Another example is shown in Figure 5 of the image of an engineering drawing form; it is important to recover the box structure of such forms, and this depends on the quality of the detection of line segments.

The pixel-wise translation symmetries can be used to construct line segments along with corners and branchpoints. A simple agglomerative scheme produces results like those

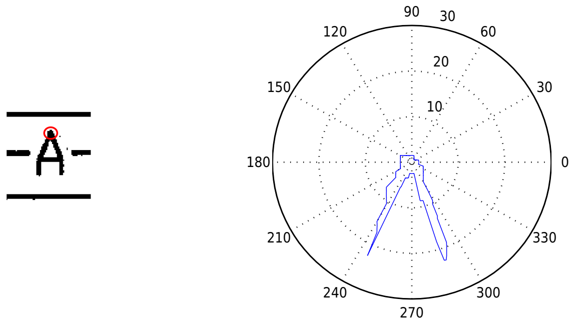

Fig. 3. Part of an engineering drawing (left) and the range scan at the indicated pixel (right).

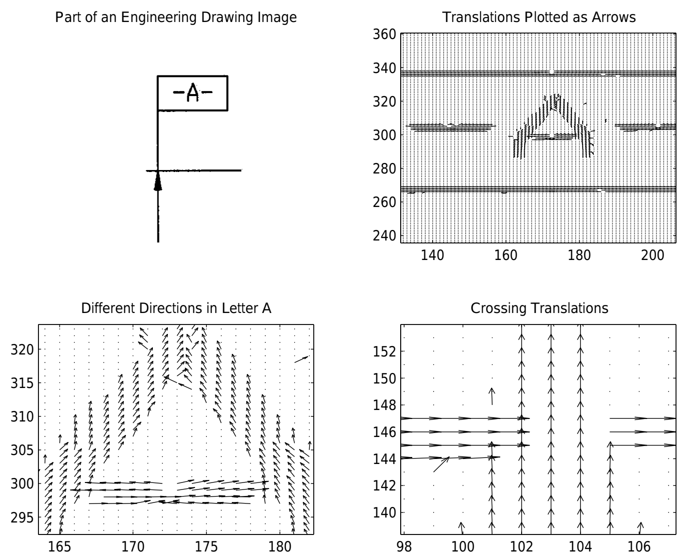

Fig. 4. Part of an engineering drawing (upper left), translation direction arrows (upper right), close-up on letter ’A’ (lower left), and close-up of crossing lines (lower right).

shown in Figure 6. The line segments are shown by orientation in different gray levels, while endpoints, branchpoints and corners are shown in red. Note that nonlinear segments give rise to many short segments. A more complete image with corners and branchpoints is shown in Figure 7. The line segments are represented as wreath products; e.g., the horizontal line segment above the letter ’A’ in the upper left part of Figure 4 is segment 61, which in Matlab is:

```
>> wp(61) -- wreath product element 61
ans =
    cg: T(179) -- 179 deg translation
    fg: [1x1 struct] -- wreath product
>> wp(61).fg
ans =
    cg: [2x2 double] -- id. transform
    fg: [114 150 113 73]  -- end pts
```

That is, a wreath product is defined recursively as a control group and a fiber group, where the fiber group is another wreath product.

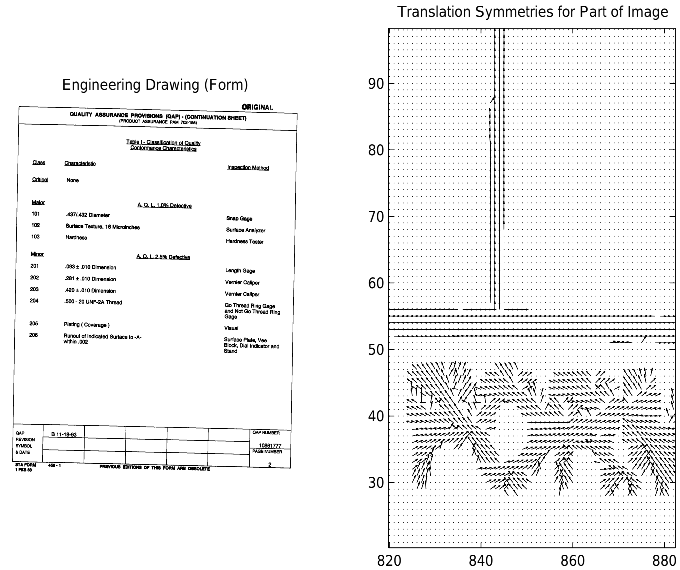

Fig. 5. Engineering forms image (left) and translation symmetries (right).

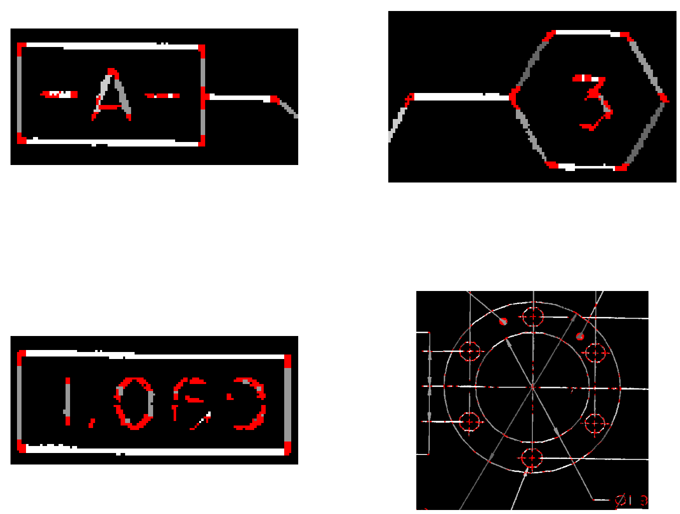

Fig. 6. A Variety of Examples of Liner Segments, Corners and Branchpoints.

B. Circles

A standard method for circle detection is the Hough transform [1], but this involves several steps which reduce the robustness of the method; however, it does work well for incomplete circles since each part of the circle contributes to its detection. Here, we focus on complete circles found in the engineering drawing. The rotation symmetry of circles can be found more easily by transforming the image into a polar form centered at a selected pixel. This can be accomplished by the *Frieze Expansion Pattern* (FEP) developed by Lee and Liu [5]. The FEP is formed by taking slices of the image in a set of directions centered at the expansion pixel. Figure 8 shows the image of a circle (left), and the corresponding FEP (right) expanded about the center of the circle.

Given the FEP from a point located inside the circle, then the significant feature about the FEP is that since the circle

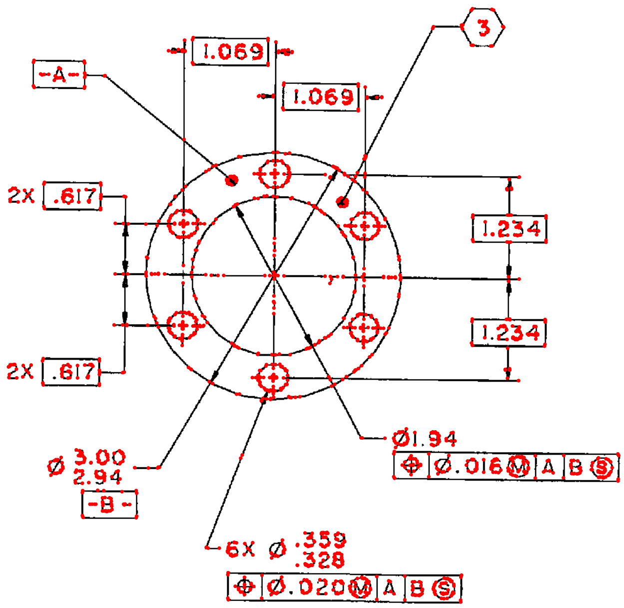

Fig. 7. Engineering Drawing showing Corners and Branchpoints).

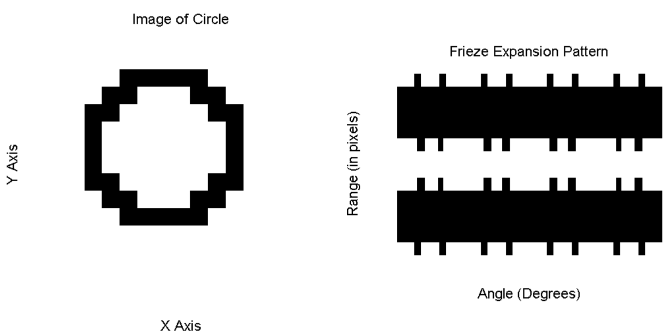

Fig. 8. Image of Circle (left) and Frieze Expansion Pattern (FEP) (right).

completely surrounds the expansion point, then there is a solid line of pixels connecting the left side of the FEP to the right side of the FEP. We take advantage of this to develop an algorithm for $O(2)$ rotation symmetry detection:

Algorithm *Find_O(2) Symmetry*

On input: im – a binary image
On output: circles – set of circles $\{x_i, y_i, r_i\}$

$\forall \bar{p}$ =(row,col)
  $FEP \leftarrow$ Frieze Expansion Pattern about $\bar{p}$
  paths_across $\leftarrow$ all distinct connected paths across FEP
  circles $\leftarrow$ paths that define circles

Consider the analysis shown in Figure 9. A complicated subimage with a pair of large circles is shown on the left. The FEP for this image (expanded about the center pixel) is shown in the middle figure, with the detected paths across the FEP from left to right. Note that there is a path across the center due to the fact that the entire middle row is formed from a foreground pixel (in this particular image); however, it has radius zero. On the right is the overlay of one of the recovered circles with the original image. Figure 10 shows a couple of more examples of $O(2)$ symmetry found in the image. Note that while the hexagon itself is a polygon, there is a subset of pixels which has a rotational symmetry. The polygonal nature of the set of points can be determined from several different aspects: (1) the translational properties of the pixels, (2) the corners, (3) the reflectional symmetries.

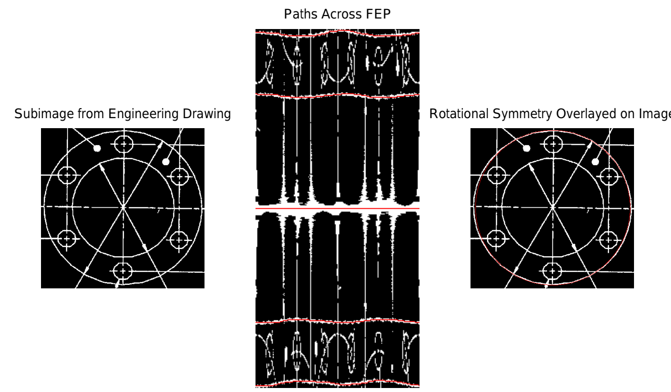

Fig. 9. Subimage with Circles (left), FEP with Overlayed Paths(middle), Recovered Rotational Symmetry overlayed on Image (right).

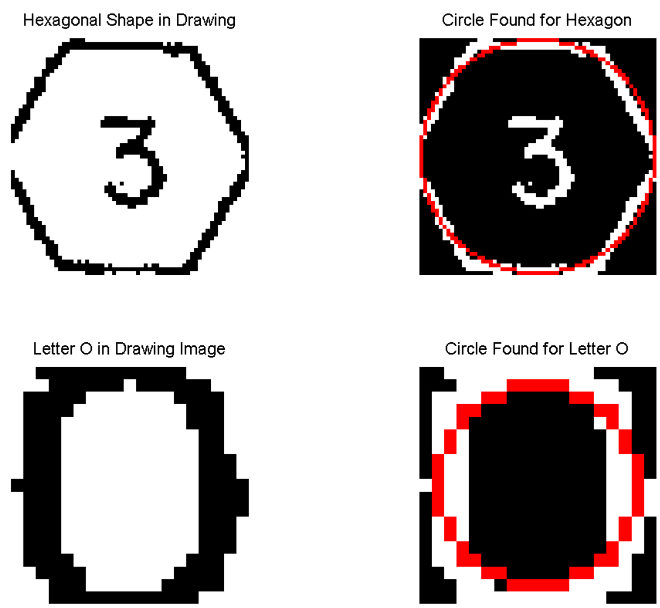

Fig. 10. Image of Hexagon (upper left) and Detected Rotational Symmetry (upper right); Image of Letter 'O' and Symmetry Detected.

C. Reflections

Reflections about a line in the plane generally require a 2D analysis of the pairwise values of points reflected across the line. Since we restrict our study to line drawings, it is possible to detect reflections by using one-dimensional curves in the FEP. The idea is that the boundary of an object (circle, polygon, etc.) exhibits the reflection symmetry by the fact that if a symmetry axis exists, then the data must have the same values going left and right from that angle in the FEP; i.e., the FEP will match the reverse of the FEP at that point. This give rise to the following algorithm:

Algorithm *Find_$\mathcal{Z}_2$ Symmetry*

On input: im – a binary image
On output: objects with reflection symmetry

$\forall \bar{p} =$(row,col)
  $FEP \leftarrow$ Frieze Expansion Pattern about $\bar{p}$
  $P_{extremal} \leftarrow$ local max and min angles in FEP
  reflection_axes $\leftarrow$ reverse FEP matches the FEP

Figure 11 shows the image of a triangle (left), its FEP (middle), and the extremal points in the FEP (right). Note that the angles associated with the extremal points (angles) are $\{15, 90, 170, 250, 270, 290\}$ degrees, and of these, only 90 and 270 satisfy the matching property. Therefore, there is a reflection axis at 90 degrees.

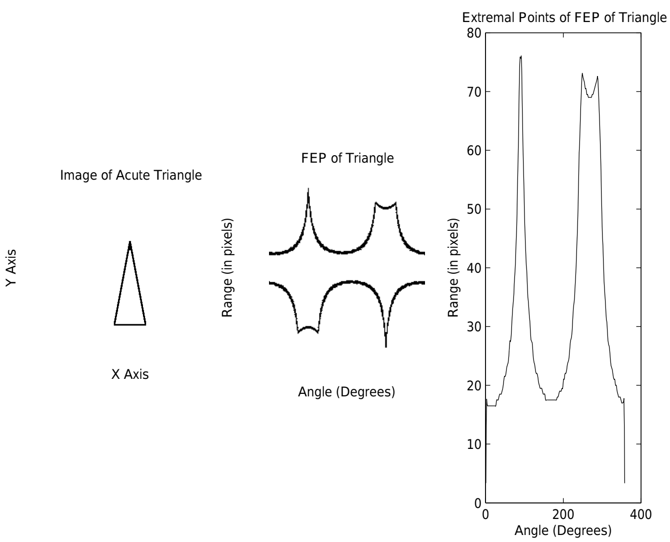

Fig. 11. Image of Acute Triangle (left), FEP of Triangle (middle), and Upper FEP Curve showing Extremal Points.

D. Rectangle Recognition

Collections of straight line segments may be examined for evidence of further relative symmetry. For example, a rectangle satisfies a 180 degree rotational symmetry, as well as two reflective symmetries through the opposite sides. We use the latter representation as the basis for analysis here. The wreath product representation is given as:

$$\{e\} \wr \Re \wr \mathcal{Z}_2 \times \mathcal{Z}_2$$

where $\mathcal{Z}_2 \times \mathcal{Z}_2$ is the direct product of the two reflection groups. This layout provides a direct method to detect rectangles. A rectangle will generally be a closed contour about the focus of expansion, and this closed contour can be checked for reflectional symmetry. (Note even if not completely closed, an A^* algorithm will be able to jump gaps.) If the number of symmetries is correct (2 reflectional), then the axes are 90 degrees apart (located as minima in the FEP curves), and at the upper and lower FEP curves are equi-distant from the center of the FEP image (see Figure 12. The method was applied to the engineering drawing image shown in Figure 13, and all the rectangles were found. An example of the rectangles found in a full engineering drawing is shown in Figure 14.

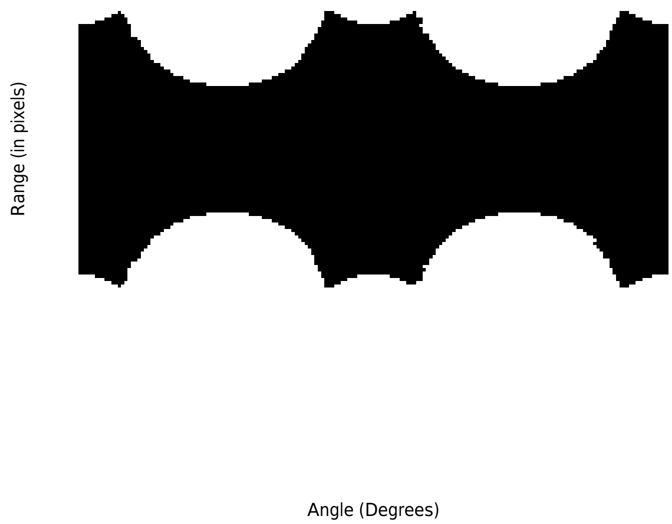

Fig. 12. FEP of Rectangle.

IV. Conclusions and Future Work

We have demonstrated a symmetry-based approach to the semantic analysis of engineering drawing images. This includes methods to discover:

1) Translational Symmetry (line segments)
2) Rotational Symmetry (circles)
3) Reflection Symmetries (Rectangles, and polygons in general)

The methods have been shown to be robust in this application.

We are currently working on several aspects to improve this approach. First, the computational cost can be reduced through the use of parallel computation (note that these methods are embarrassingly parallel – i.e., can be run in parallel at each pixel in the image). In addition, we plan to extend the method to the analysis of text. Most characters and digits have some intrinsic symmetry which can be used to achieve better recognition rates. Finally, we are currently applying these techniques to a large collection of engineering drawings and investigating the use of machine learning methods to further improve performance.

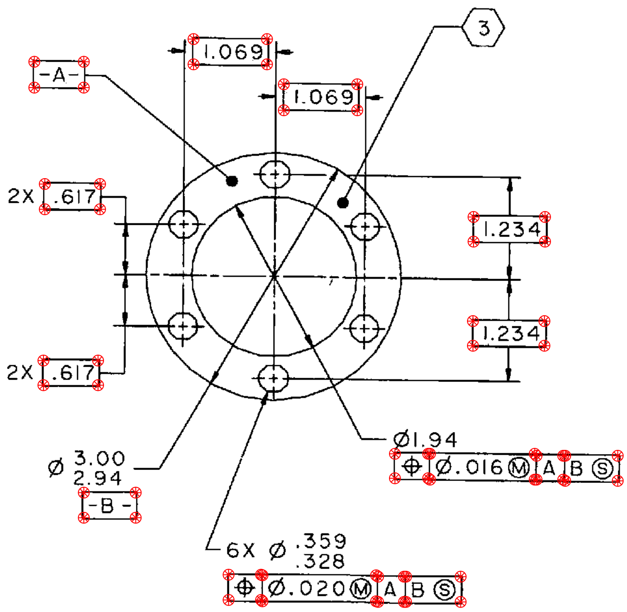

Fig. 13. Rectangles found in Engineering Drawing.

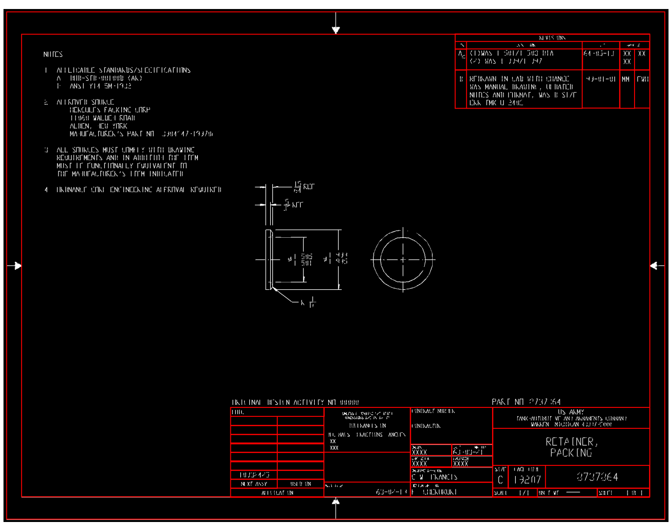

Fig. 14. Rectangles found in Full Engineering Drawing.

Acknowledgments: This work was supported by in part by AFOSR-FA9550-12-1-0291.

REFERENCES

[1] D.A. Forsyth and J. Ponce. *Computer Vision*. Prentice Hall, Upper Saddle River, NJ, 2003.

[2] T.C. Henderson. *Analysis of Engineering Drawings and Raster map Images*. Springer Verlag, New York, NY, 2014.

[3] Thomas C. Henderson and Lavanya Swaminathan. NDAS: The Nondeterministic Agent System for Engineering Drawing Analysis. In *Proceedings of the International Conference on Integration of Knowledge Intensive Multi-agent Systems*, pages 512–516, Boston, MA, October 2003.

[4] Anshul Joshi, Thomas C. Henderson, and Wenyi Wang. Robot Cognition using Bayesian Symmetry Networks. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, Angers, France, March 2014.

[5] S. Lee and X. Liu. Skewed Rotation Symmetry Group Detection. *IEEE PAMI*, 32(9):1659–1672, 2009.

[6] M. Leyton. *A Generative Theory of Shape*. Springer, Berlin, 2001.

[7] M.A.G. Viana. *Symmetry Studies*. Cambridge University Press, Cambridge, UK, 2008.

[8] H. Weyl. *Symmetry*. Princeton University Press, Princeton, NJ, 1952.